School Attendance Scraper

Web scraper for collecting teacher attendance data from Delhi Education Department's portal. (without selenium web driver)

What It Does

3-step pipeline:

1. **Fetch HTML** - Gets data from endpoint with parameters (date, name, attrtype)
2. **Parse Data** - Extracts employee attendance from HTML tables
3. **Save to Excel** - Outputs clean Excel file

Attendance types: Present, On Duty, half casual leave, vacation, etc

Output columns: school_id, school_name, employee_id, employee_name, attendance_status

Quick Start

1. **Install dependencies:**

```
pip install -r requirements.txt
```

2. **Edit `config.py` to add your schools:**

```
DAT = '10'  # Day of month (1-31)

SCHOOLS = [
    '1002403-Govt. Coed Secondary School,Joshi colony,Mandawali',
    '1002404-Another School Name',
    # Add more schools here
]
```

3. **Run the pipeline:**

```
python main.py
```

4. **Check output:** Files saved as `data/schoolcode_date.xlsx`

Parameters

Edit `config.py` to configure:

- **DAT**: Day of month (1-31)
- **SCHOOLS**: List of schools in format "code-name"
- **ATTENDANCE_TYPES**: Which attendance types to scrape
- **OUTPUT_DIR**: Output directory (default: 'data')

Output

Creates separate Excel files for each school named: `school-name_date.xlsx`

Each file contains columns:

- school_id
- school_name
- employee_id
- employee_name
- attendance_status

Advantages of This Approach

Selenium would need much more navigation through browser interactions. This approach offers:

- **No Browser Overhead:** Much faster than Selenium
- **Lightweight:** Minimal dependencies
- **Easy to Deploy:** Can be easily scheduled on Great Lakes
- **Efficient:** Can process hundreds of schools quickly
- **Maintainable:** Clean separation of concerns (**fetch, parse** and **save data**)
- **Testable:** Easy to unit test individual components

Project Structure

```
Final_Task/
├── pipeline/
│   ├── fetch_html.py          # Fetch HTML from endpoint
│   ├── parse_attendance.py    # Parse attendance data
│   └── save_data.py           # Save to Excel
├── jobs/
│   └── job_scheduler.slurm    # SLURM script (will be implemented as
required)
├── data/
│   └── schoolcode_date.xlsx    # Output Excel files
├── config.py                   # Configuration (set schools, dates)
├── main.py                     # Main pipeline script
├── requirements.txt            # Python dependencies
└── README.md                   # This file
```

Dependencies

- `requests` - HTTP requests
- `beautifulsoup4` - HTML parsing
- `pandas` - Data handling
- `openpyxl` - Excel output